Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A method of executing a plurality of tasks of different priority values, said method comprising:

enabling any task irrespective of priority value to request a particular waiting period during execution of said task, wherein said task requesting said particular waiting period occurs irrespective of elapsed time due to execution of said task;

utilizing preemptive multitasking to enable each task to preempt execution of another task based on said requested particular waiting period elapsing, said priority values, and a plurality of statuses associated with said tasks; and

utilizing cooperative multitasking to enable each task to suspend execution of itself in a cooperative manner for duration of said requested particular waiting period, wherein said preemptive multitasking and said cooperative multitasking increase utilization of processing power of a processor and ensure higher priority valued tasks are executed with less interruption time than lower priority valued tasks.

2. (Currently Amended) The method as recited in Claim 1 further comprising:

selecting a task from said tasks based on said priority values and a said plurality of statuses associated with said tasks, wherein said statuses include executing, waiting, interrupted, completed, and unstarted;

starting said selected task and designating said selected task an executing task;

if said executing task requests a waiting period, suspending said executing task and designating said executing task a waiting task and repeating said selecting said task and said starting said selected task;

if said waiting period elapses for any waiting task and said executing task has a higher priority value than said waiting task, designating said waiting task an interrupted task;

if said waiting period elapses for any waiting task and said executing task does not have a higher priority value than said waiting task, suspending said executing task and designating said executing task an interrupted task and repeating said selecting said task and said starting said selected task; and

if said executing task completes execution, designating said executing task a completed task and repeating said selecting said task and said starting said selected task.

3. (Original) The method as recited in Claim 2 wherein said selecting said task includes:

selecting higher priority values before selecting lower priority values when possible.

4. (Original) The method as recited in Claim 2 wherein said selecting said task includes:

if a first particular task cannot be executed until a second particular task has completed execution, enabling selection of said first particular task after said second particular task has completed execution.

5. (Original) The method as recited in Claim 2 further comprising:

setting a timer based on said waiting period.

6. (Original) The method as recited in Claim 1 wherein said tasks are BIOS (Basic Input Output System) initialization tasks.

7. (Original) The method as recited in Claim 6 wherein said waiting period is requested from a BIOS kernel.

8. (Currently Amended) A computer-readable medium comprising computer-executable instructions stored therein for performing a method of executing a plurality of tasks of different priority values, said method comprising:

enabling any task irrespective of priority value to request a particular waiting period during execution of said task, wherein said task requesting said particular waiting period occurs irrespective of elapsed time due to execution of said task;

utilizing preemptive multitasking to enable each task to preempt execution of another task based on said requested particular waiting period elapsing, said priority values, and a plurality of statuses associated with said tasks; and

utilizing cooperative multitasking to enable each task to suspend execution of itself in a cooperative manner for duration of said requested particular waiting period, wherein said preemptive multitasking and said cooperative multitasking increase utilization of processing power of a processor

and ensure higher priority valued tasks are executed with less interruption time than lower priority valued tasks.

9. (Currently Amended) The computer-readable medium as recited in Claim 8 wherein said method further comprises:

selecting a task from said tasks based on said priority values and a said plurality of statuses associated with said tasks, wherein said statuses include executing, waiting, interrupted, completed, and unstarted;

starting said selected task and designating said selected task an executing task;

if said executing task requests a waiting period, suspending said executing task and designating said executing task a waiting task and repeating said selecting said task and said starting said selected task;

if said waiting period elapses for any waiting task and said executing task has a higher priority value than said waiting task, designating said waiting task an interrupted task;

if said waiting period elapses for any waiting task and said executing task does not have a higher priority value than said waiting task, suspending said executing task and designating said executing task an interrupted task and repeating said selecting said task and said starting said selected task; and

if said executing task completes execution, designating said executing task a completed task and repeating said selecting said task and said starting said selected task.

10. (Original) The computer-readable medium as recited in Claim 9 wherein said selecting said task includes:

selecting higher priority values before selecting lower priority values when possible.

11. (Original) The computer-readable medium as recited in Claim 9 wherein said selecting said task includes:

if a first particular task cannot be executed until a second particular task has completed execution, enabling selection of said first particular task after said second particular task has completed execution.

12. (Original) The computer-readable medium as recited in Claim 9 further comprising:

setting a timer based on said waiting period.

13. (Original) The computer-readable medium as recited in Claim 8 wherein said tasks are BIOS (Basic Input Output System) initialization tasks.

14. (Original) The computer-readable medium as recited in Claim 13 wherein said waiting period is requested from a BIOS kernel.

15. (Currently Amended) A system comprising:

a processor; and

a BIOS (Basic Input Output System) operative to utilize preemptive multitasking and cooperative multitasking to increase utilization of processing power of said processor and to ensure higher priority valued initialization tasks are executed with less interruption time than lower priority valued initialization tasks when executing a plurality of initialization tasks of different priority values, wherein said BIOS is operative to enable any initialization task irrespective of

priority value to request a particular waiting period during execution of said initialization task, ~~and~~ wherein said initialization task requesting said particular waiting period occurs irrespective of elapsed time due to execution of said initialization task, wherein said BIOS uses said preemptive multitasking to enable each initialization task to preempt execution of another initialization task based on said requested particular waiting period elapsing, said priority values, and a plurality of statuses associated with said initialization tasks, and wherein said BIOS uses said cooperative multitasking to enable each initialization task to suspend execution of itself in a cooperative manner for duration of said requested particular waiting period.

16. (Original) The system as recited in Claim 15 wherein when executing said initialization tasks, said BIOS selects an initialization task having a higher priority value before selecting an initialization task having a lower priority value when possible.

17. (Original) The system as recited in Claim 15 wherein if a first particular initialization task cannot be executed until a second particular initialization task has completed execution, said BIOS enables selection of said first particular initialization task after said second particular initialization task has completed execution.

18. (Original) The system as recited in Claim 15 further comprising:

a timer.

19. (Original) The system as recited in Claim 15 wherein said BIOS includes a BIOS kernel for receiving requests for said particular waiting period from said initialization tasks.

20. (Original) The system as recited in Claim 15 further comprising a plurality of hardware components.